

Klausur

Computergrundlagen WS 2015/2016

Dr. Maria Fyta Dr. Jens Smiatek Tobias Richter
Bibek Adhikari Julian Michalowsky Jonas Landsgesell
Frank Maier Robin Bardakcioglu Alexander Weyman
Nicolai Roth

1. März 2016

Name	
Vorname	
Matrikelnummer	

Hinweise

- Die Maximalpunktzahl ist 100.
- Der verfügbare freie Platz gibt einen Hinweis darauf, welchen Umfang die Lösung haben sollte.
- Die Klausur ist zu umfangreich um alle Themengebiete abdecken zu können. In der Regel wird es nicht möglich sein, alle Aufgaben vollständig zu bearbeiten. Bearbeiten Sie deswegen zuerst die Themengebiete, die Ihnen besonders liegen!
- Falls der Platz nicht ausreichen sollte, verwenden Sie zusätzliche Blätter. Beschriften Sie diese unbedingt mit Ihrem Namen und Ihrer Matrikelnummer!
- Einige Fragen ähneln den Fragen aus vorigen Klausuren, sind aber *nicht* identisch! Lesen Sie die Fragen deshalb bitte *genau* durch!

Viel Erfolg!

1 Unixgrundlagen (25 Punkte)

Aufgabe 1: (1 Punkt)

Was ist der Unterschied zwischen einem Terminal und einer Shell?

Antwort:

- * Ein Terminal ist das angezeigte Fenster.
- * Es stellt die Schnittstelle zwischen Benutzer und Shell dar.
- * In einem Terminal läuft eine Shell.
- * Die Shell führt die eingegebenen Befehle aus.

Aufgabe 2: (1 Punkt)

Was ist der Unterschied zwischen dem World Wide Web und dem Internet?

Antwort:

Was ist der Unterschied zwischen dem WWW und dem Internet? Das Internet ist ein Verbund aller Netzwerke, die mittels des Internet-Protokolls (IP) kommunizieren können. Das WWW ist ein Dienst auf dem Internet, der es erlaubt, über das Hypertext-Transfer Protocol (HTTP) Hypertextdaten auszutauschen.

Aufgabe 3: (1 Punkt)

Ist ssh ein Dienst des Internets oder des World Wide Webs?

Antwort:

ssh ist ein Dienst des Internets

Aufgabe 4: (1 Punkt)

Wie wird die Hilfeseite von grep aufgerufen?

Antwort:

man grep

Aufgabe 5: (2 Punkte)

Was macht der folgende Shell-Befehl? Erklären Sie nicht die einzelnen Kommandos, sondern die Gesamtfunktion.

```
ssh cip0 "cat_/proc/cpuinfo" > ~/cip0version.txt
```

Antwort:

Es wird der Inhalt der Datei /proc/cpuinfo des Rechners cip0 im Terminal über ssh ausgelesen und in die Datei cip0version.txt im lokalen Heimatverzeichnis geschrieben.

Aufgabe 6: (2 Punkte)

Benutzen Sie die Programme `grep` und `wc` um in der Datei `loremipsum.txt` das Vorkommen der Zeichenkette `et` zu zählen. Groß- und Kleinschreibung sollen dabei keine Rolle spielen.

Antwort:

```
cat loremipsum.txt | grep -i et | wc -l  
oder  
grep -i et loremipsum.txt | wc -l
```

Aufgabe 7: (2 Punkte)

Geben Sie den Befehl an, der alle Dateien mit der Endung `.dat` im aktuellen Verzeichnis ausgibt. Dabei sollen auch die versteckten Dateien und Verzeichnisrechte ausgegeben werden.

Antwort:

```
ls -la *.dat
```

Aufgabe 8: (4 Punkte)

Markieren und kategorisieren (Syntax-, Logik-Fehler) Sie die vier Fehler im folgenden bash-Skript.

```
1 #!/bin/bash  
2  
3 myDir=$1  
4 fileList=$(ls $myDir/*.txt) #Leerzeichen vor =  
5 today=$(date +%y-%m-%d)  
6  
7 for oldFile in $fileList; do #falscher Variablenname fileList  
8     basename=$(basename $oldFile .txt) #oldFile anstatt $oldFile  
9     newFile=$myDir/${basename}_${today}.txt  
10    cp $oldFile $newFile && rm $oldFile  
11    echo "moved_$oldFile_to_$newFile" >> doSomething.log  
12 done #fi anstatt done
```

Aufgabe 9:

(2 Punkte)

Was macht das obige Skript *nachdem* die Fehler korrigiert wurden?

Antwort:

Es nimmt einen Ordner als Command-Line-Argument an.
Alle in diesem Ordner befindlichen Dateien, die die Endung `.txt` tragen bekommen die Endung `_Datum.txt` mit dem heutigen Datum
Es wird ein Logfile mit den vorgenommenen Änderungen erstellt.

Aufgabe 10:

(1 Punkt)

Was wird im Terminal, *nachdem* die Fehler korrigiert wurden, ausgegeben?

Antwort:

Es wird nichts ausgegeben.

Aufgabe 11:

(1 Punkt)

Wie kann Zeile 10, aus dem obigen Skript, kürzer geschrieben werden?

Antwort:

`mv $oldFile $newFile`

Aufgabe 12:

(1 Punkt)

Wie kann das obige Skript `doSomething.sh` im Terminal aufgerufen werden, so dass die Aufgabe im Ordner `./test` ausgeführt wird, auch wenn das Skript im Arbeitsverzeichnis `./` liegt?

Antwort:

`./doSomething.sh test`

An einem Institutscomputer hat Benutzer `richter` folgenden Dialog in der Shell:

```
> groups richter weeber cgl15-001
richter : icp klausur cgl video pc
weeber  : fluid pc icp granular guest sysguru video
cgl15-001 : cgl
> ls -la
total 8
drwxr-xrwx 5 richter cgl  76 Oct 29 13:55 .
drwxr-xr-x 3 richter icp  17 Oct 29 12:01 ..
-rw-r----- 1 richter cgl 441 Oct 29 12:01 bar.txt
dr-xrwxr-x  2 richter cgl   6 Oct 29 12:01 cgl2014
----rw----  1 richter cgl 260 Oct 29 12:01 foo.txt
```

Aufgabe 13:

(3 Punkte)

Welcher der Benutzer `richter`, `weeber` und `cgl15-001` kann welchen der folgenden Befehle erfolgreich ausführen?

```
cat bar.txt >> newfile.txt
mv cgl2014 cgl2015
```

Antwort:

cat... read bar: richter, cgl15; write ./: richter, weeber. => darf nur richter
mv... write ./: richter, weeber

Aufgabe 14:

(3 Punkte)

Welchen Befehl muss `richter` ausführen, damit Benutzer `cgl15-001` einen neuen Ordner im aktuellen Verzeichnis anlegen kann?

Antwort:

chmod g+w . oder chmod a+w . oder chmod 777 . oder ...

2 Python (20 Punkte)

Aufgabe 15:

(5 Punkte)

Schreiben Sie eine Python-Funktion $\ln(x, n)$, die mittels einer Taylor-Expansion den natürlichen Logarithmus von $1 + x$ bis zum n -ten Glied berechnet.

Hinweis: $\ln(1 + x) = \sum_{i=1}^n \frac{(-1)^{i-1}}{i} x^i$

Antwort:

```
def ln(x, n):
    result = 0
    for i in range(n):
        result += (-1)**i / (i+1.0) * x**(i+1.0)
    return result
```

Aufgabe 16:

(2 Punkte)

Was müssen Sie ausführen, um die Funktionen des numpy Pakets in Ihrem Python-Skript verwenden zu können?

Antwort:

Am Beginn sollte irgendwas derart:

```
import numpy
import numpy as np
from numpy import *
```

stehen

Aufgabe 17:

(3 Punkte)

Inwiefern unterscheidet sich Python grundsätzlich von kompilierten Sprachen wie zum Beispiel Fortran oder Ada? Nennen Sie zusätzlich je einen Vorteil für beide Konzepte.

Antwort:

Interpreter Sprache, Skripte werden Zeile für Zeile ausgeführt. Ein externes Programm, der Interpreter, ist notwendig um ein "fertiges" Programm auszuführen. Ein kompiliertes Programm kann ohne Zusätze auf allen (kompatiblen) Computern benutzt werden.

Vorteil Skript: Einfach zu debuggen, erstellen Vorteil Comp: Einfach schneller

Aufgabe 18:

(4 Punkte)

Was ist die Aufgabe der folgende Funktion? Wie könnte man die Funktion signifikant beschleunigen (falls die verwendeten Vektoren hochdimensional/lang sind)?

```
import numpy as np

length=100
vec_a=100*np.random.random(length)
vec_b=100*np.random.random(length)

def prod(vec_a,vec_b):
    prod=0
    for i in range(len(vec_a)):
        prod+=vec_a[i]*vec_b[i]
    return prod

print prod(vec_a,vec_b)
```

Antwort:

Die Funktion berechnet ein Skalarprodukt. Statt das Skalarprodukt in Python zu berechnen, kann man es auch in numpy berechnen (mit `np.dot(a,b)`), gerade bei Vektoren in hohen Dimensionen wird numpy aufgrund der Vektorisierung schneller sein.

Aufgabe 19:

(6 Punkte)

Das folgende Python Skript berechnet die Nullstellen eines Polynoms dritten Grades mithilfe des Newton-Verfahrens, enthält allerdings einige Fehler. Markieren und erklären Sie die Fehler.

```
def g(x):
    return x^3+3*x-9*x-2

def newton(function, x0)
    double dx= 0.001
    num_deriv_at_x0=(function(x0)-function(x0+dx))/-dx
    xn=x0-function(x0)/num_deriv_at_x0
    if(abs(x0-xn)<0.00001):
    then
        return xn
    else:
        return newton(function,xn)
    fi

echo newton(g,0), newton(g,3), newton(g,-3)
```

Antwort:

```
def g(x):
    return x**3+3*x-9*x-2 // ** muss verwendet werden

def newton(function, x0): //Doppelpunkt fehlt
    dx=0.001 //keine Deklaration moeglich
    num_deriv_at_x0=(function(x0)-function(x0+dx)) / (-dx)
    xn=x0-function(x0)/num_deriv_at_x0
    if(abs(x0-xn)<0.00001)://kein then
        return xn
    else:
        return newton(function,xn)
    //kein fi

print newton(g,0), newton(g,3), newton(g,-3) //es gibt kein echo in p
```


3 C (10 Punkte)

Aufgabe 20:

(5 Punkte)

Das folgende Programm in C enthält leider drei Fehler. Korrigieren Sie die Fehler und erläutern Sie die Aufgabe des Programms.

```
include<stdio.h>

double com(int N, double r, double* m) {
    double total_mass=0;
    double r_com=0.0;
    for(int i=0;i>N;i++){
        r_com+=r[i]*m[i];
        total_mass+=m[i];
    }
    return r_com/total_mass;
}

int main() {
    double r[5]={1,2,3,4,5};
    double m[5]={1,2,1,1,2};
    int number_of_particles=5;
    double r_com=com(number_of_particles,r,m);
    printf("Com_is_%f\n",r_com);
}
```

Antwort:

Das Programm berechnet den Massenswerpunkt in einer dimension

```
#include<stdio.h> // missing #

double com(int N, double* r, double* m) {
// missing * for position array r
    double total_mass=0;
    double r_com=0.0;
    for(int i=0;i<N;i++){ // mixed up > and <
        r_com+=r[i]*m[i];
        total_mass+=m[i];
    }
    return r_com/total_mass;
}

int main() {
    double r[5]={1,2,3,4,5};
    double m[5]={1,2,1,1,2};
    int number_of_particles=5;
    double r_com=com(number_of_particles,r,m);
    printf("Com_is_%f\n",r_com);
}
```

Aufgabe 21:

(5 Punkte)

- Erklären Sie in einem kurzen Satz, was ein Pointer ist. (1 Punkt)
- Erklären Sie den Zweck, bzw. Nutzen eines Compilers. (2 Punkte)
- Mit welchem Befehl wird aus der C-Datei `beispiel.c` ein ausführbares Programm namens `beispiel` erzeugt? (1 Punkt)
- Welchen Wert hat i nach der Ausführung der letzten gezeigten Zeile (siehe unten)? (1 Punkt)

```
...
int i = 5;
i++;
i -= 2;
...
```

Antwort:

- Entweder: Ein Pointer ist eine Speicheradresse. Oder: Ein Pointer zeigt auf den Ort im Speicher, in dem sich die referenzierten Daten befinden.
- Ein Compiler Übersetzt Code in Maschinsprache/in eine ausführbare Datei. In einigen Programmiersprachen kann man den Code nur dann ausführen, wenn dieser vorher compiled wurde.
- `cc beispiel.c -o beispiel`
- $i = 4$

4 Algorithmen und Datenstrukturen (20 Punkte)

Aufgabe 22:

(6 Punkte)

Unten ist ein Programmausschnitt angegeben, der die grobe Struktur des Radixsort-Algorithmus' in Pseudocode darstellt. In diesem Fall sollen die Binärzahlen, die in `a` gespeichert sind, sortiert werden.

```
a=[101, 10, 111, 0, 100]
for(stelle=1..n){
    listen = auf_Listen_verteilen(stelle,a)
    a = Listen_zusammenfuegen(listen)
}
```

- Wie oft wird die for-Schleife in diesem Beispiel durchlaufen werden, d.h. welchen Wert muss `n` haben? (1 Punkt)
- Die Variable `listen` soll eine Liste von Listen von Binärzahlen sein. Wie viele Listen muss sie enthalten? (1 Punkt)
- Führen Sie nun den Algorithmus von Hand aus. Geben Sie hierfür für jeden Schleifendurchlauf an, wie die Listen `listen[0]` etc., nach dem Funktionsaufruf von `auf_Listen_verteilen()` und `a` nach dem Funktionsaufruf von `Listen_zusammenfuegen()` aussehen. (4 Punkte)

Antwort:

- 3 = maximale Stellen in `a`
- 2 da Binärzahlen sortiert werden sollen
- `stelle=1`: `listen[0]=[10,0,100]`, `listen[1]=[101,111]`, `a=[10,0,100,101,111]`
`stelle=2`: `listen[0]=[0,100,101]`, `listen[1]=[10,111]`, `a=[0,100,101,10,111]`
`stelle=3`: `listen[0]=[0,10]`, `listen[1]=[100,101,111]`, `a=[0,10,100,101,111]`
(0.5 Punkte pro (richtige Liste - 1))

Aufgabe 23:

(2 Punkte)

Was ist ein Array?

Antwort:

Ein Array ist eine Datenstruktur, die viele Elemente desselben Datentyps enthält, wobei jedem Element ein Index zugeordnet ist.

Aufgabe 24:

(12 Punkte)

«««< HEAD Schreibe ein Programm, das durch das Ziehen von Zufallszahlen (Monte-Carlo Integration) eine Approximation für π berechnet und ausgibt. Das Programm kann in einer Programmiersprache deiner Wahl oder auch Pseudocode geschrieben werden. Verwende dabei eine Funktion `random()`, die nicht importiert werden muss und gleichverteilte Zufallszahlen zwischen 0 und 1 erzeugt. Syntaxfehler sind irrelevant, solange klar ist, was das Programm tut. ===== Schreiben Sie ein Programm, das durch das Ziehen von Zufallszahlen (Monte-Carlo Integration) eine Approximation für π berechnet und ausgibt. Das Programm kann in einer Programmiersprache Ihrer Wahl oder auch Pseudocode geschrieben werden. Verwenden Sie dabei eine Funktion `random()`, die nicht importiert werden muss und gleichverteilte Zufallszahlen zwischen 0 und 1 erzeugt. Syntaxfehler sind irrelevant, solange der Ablauf des Programmes eindeutig ist.

Antwort:

```
max_steps=1000000      # (1P)
sum=0                  # (1P)
for (i=1:max_steps) {  # (1P)
    x=random()         # (0.5P)
    y=random()         # (0.5P)
    if (x**2+y**2 < 1) {      # (3P)
        sum++           # (1P)
    }
}
erg=4*sum/max_steps   # (3P)
print(erg)            # (1P)
```

5 LaTeX (15 Punkte)

Aufgabe 25:

(5 Punkte)

```
\begin{document}

{\huge\textbf{1. Einkaufsliste}}\
{\large{1.1. Supermarkt}}\
1. Milch
2. Mehl
3. Eier

\textbf{Wichtig} R"omer k"onnen warten,
Wildschweine nicht!

\end{document}
```

Obiger L^AT_EX Quellcode verwendet visuelles Markup. *Benennen* Sie die verwendeten visuellen Markup Befehle und *notieren* Sie dazu logische Markup-Befehle mit denen Sie die visuellen ersetzen könnten.

Antwort:

0.5 Punkte: section, 0.5 Punkte Umsetzung
0.5 Punkt: subsection, 0.5 Punkte Umsetzung
1 Punkt: enumerate, 0.5 Punkte \begin{enumerate}, 0.5 Punkte \item \$
0.5 Punkte paragraph, 0.5 Punkte \paragraph{Wichtig} \$\ldots\$

Aufgabe 26:

(5 Punkte)

Notiere den L^AT_EX Quellcode mit dem Sie die Formel (1) mit Nummerierung erstellen können.

$$f_k^{\text{test}} = \frac{\alpha \cdot e^{-2x}}{2k} \text{ für } k > 0 \quad (1)$$

Antwort:

- | 0.5 + 0.5 Punkte: `\text{}`
- | 1 Punkt: `\frac{}`
- | 0.5+0.5 Punkt: `\,`
- | 0.5 Punkt: `\alpha`
- | 0.5 Punkt: `\cdot`
- | 0.5 Punkt: `^{-2x}`
- | 0.5 Punkt: `f_k^`

Aufgabe 27:

(5 Punkte)

Notieren Sie einen L^AT_EX Quellcode, mit der die abgebildete Tabelle erstellt werden kann.

Zeit [s]	Strecke [m]	Geschwindigkeit [$\frac{m}{s}$]
0	10	2
1	12	2
2	14	2

Antwort:

```

1 Punkt \begin{tabular}{}\end{tabular}
1 Punkt {r|cc}
1 Punkt \hline
0.5 Punkte $\frac{m}{s}$, $$, $m$,
1 Punkt &&
0.5 Punkte \\
    
```

6 Visualisierung (10 Punkte)

Aufgabe 28:

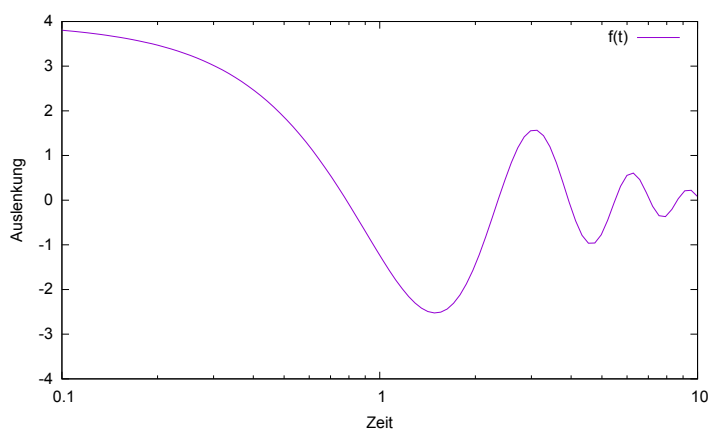
(2 Punkte)

Geben Sie an welches Dateiformat für die folgenden Einsatzzwecke gewählt werden kann:

- Photo mit dem Smartphone aufnehmen
- Einfaches LaTeX Logo, welches auf einem Werbeplakat gedruckt werden soll
- Graph von Messwerten
- Auf dem Bildschirm dreidimensional visualisiertes Protein

Aufgabe 29:

(5 Punkte)



Abgebildet ist der Graph der Funktion f mit dem Funktionsterm

$$f(t) = 4e^{-0.3t} \cos(2t)$$

Erstellen Sie ein vollständiges Gnuplot-Skript, welches ein PDF Dokument mit dem Namen 'plot.pdf' erstellt und den Graphen von f wie oben abgebildet, enthält.

Antwort:

```
#0.5 Punkte
set out "plot.pdf"
#0.5 Punkte
set term pdf

#0.5 Punkte
set xlabel "Zeit"
#0.5 Punkte
set ylabel "Auslenkung"

#0.5 Punkte
set xrange [0.1:10]
#0.5 Punkte
set yrange [-4:4]

# 0.5 Punkte
set logscale x

# 1.5 Punkte (0.5 für Titel, 0.5 für x anstatt t, 0.5 für die Formel)
plot 4*exp(-0.3*x)*cos(2*x) title 'f(x)'
```


Aufgabe 30:

(3 Punkte)

Skizzieren Sie das mit folgendem Python-Skript erstellte Schaubild.

```
from numpy import *
from matplotlib.pyplot import *
x = np.arange(0., 10., 0.1)
y = np.cos(x)
subplot(121)
plot(x, x, '-', x, x**2, 'o', x, x**3, '+')
subplot(122)
plot(x, y, lw=2)
ylim(-1,1)
show()
```

Antwort:

1.5 Punkt: subplot121
1.5 Punkt: subplot122